

Installation und Konfiguration von FreeBSD auf einem STRATO-Server mit Hilfe des Depenguinators

Christian Reiss
Clemens Renner

<http://strato.alpha-labs.net/>

29. Mai 2006

Inhaltsverzeichnis

1	Einleitung	5
1.1	Rechtliches	5
1.2	Vorwort	5
1.3	Voraussetzungen	6
2	Vorbereitung	7
2.1	Sammeln von Informationen	7
2.2	Konfiguration des Depenguinators	7
2.3	Erstellen des Depenguinator-Images	9
3	Arbeiten mit dem Depenguinator	11
3.1	Konzept	11
3.2	Partitionieren (Rescue Modus) (Linux)	11
3.3	Installation des Depenguinators	12
3.4	Alternative Methode	13
3.5	Noch eine Methode	14
3.6	Neu starten	14
4	Unter FreeBSD	17
4.1	Netzwerk (vorerst) einrichten	17
4.2	Partitionieren	17
4.3	Konfiguration von sysinstall	19
4.4	Installation	19
5	Konfiguration des installierten Systems	21
5.1	chroot	21
5.2	Sendmail	21
5.3	/etc/rc.conf	21
5.4	/etc/fstab	22
5.5	/etc/ttys	22
5.6	/etc/sysctl.conf	23
5.7	/etc/resolv.conf	23
5.8	Root-Passwort	23
5.9	Notfall-Hintertür	23
5.10	Neustart	24

6	Das neue System	25
6.1	cvsup	25
6.2	/boot.config	26
6.3	World, Kernel und Loader neu bauen	26
6.4	Der letzte Neustart	27
6.5	Backdoor entfernen	27
6.6	Letzte Worte	27
A	Frequently Asked Questions	29
B	Konfigurationsdateien	31
B.1	initconf.conf(.dist) (dp)	31
B.2	initconf (dp)	32
B.3	/etc/cvsup	33
B.4	/etc/cvsup-ports	33
B.5	/etc/make.conf	34
B.6	/etc/ttys	34
B.7	/etc/sysctl.conf	34
B.8	/etc/fstab	34

1 Einleitung

1.1 Rechtliches

Das vorliegende How-To ist rein informativ. Wer dieser Anleitung folgt, tut dies auf seine eigene Verantwortung. Die Autoren übernehmen keinerlei Haftung für eventuelle Schäden, die durch die Ausführung beschriebener Schritte entstehen können. Das Dokument darf als Ganzes frei verteilt werden, solange es unmodifiziert bleibt.

Auf inhaltliche Korrektheit wurde großen Wert gelegt. Wer Verbesserungsvorschläge hat oder einen Fehler entdeckt, wird gebeten, uns eine E-Mail zukommen zu lassen.

Wir freuen uns auch über Erfahrungsberichte, wenn es geklappt hat!

1.2 Vorwort

Heutzutage gibt es dedizierte Server an jeder Straßenecke. Die meisten davon laufen unter SuSE oder einer anderen Geschmacksrichtung von Linux. Eine Handvoll Anbieter hat auch eine serielle Konsole spendiert, die förmlich danach schreit, dass man ein anderes Betriebssystem installiert.

Mit dem richtigen Wissen, geeignetem Werkzeug und etwas Handarbeit kann man aus einem SuSE-Spielzeug einen richtigen Server basteln. Doch bis dahin ist es ein weiter Weg, denn das Rescue-System kennt kein UFS, und somit ist ein Zugreifen auf die FreeBSD-Slices nicht möglich. Ein einfaches Hochladen und Entpacken vorgefertigter TAR-Dateien fällt damit leider aus.

Deshalb ist es wichtig, vorher alle Informationen zu sammeln und diese gezielt in einem Abwasch korrekt einzutragen. Selbst der kleinste Fehler ist meist schon eine Freifahrkarte zum Neubeginn.

Die Autoren dieses Dokuments haben am eigenen Server jeden einzelnen Fehler zu spüren bekommen und wissen daher, wo sich die Knackpunkte verstecken.

Mit dieser Anleitung hoffen wir, vor Fehlern bewahren zu können, und wünschen gleichzeitig eine reibungslose Installation.

Die Motivation für die Autoren, dieses Dokument zu schreiben, war in erster Linie, dass es kein vergleichbares Dokument gibt¹. Hinweise und Tipps existieren auf vielen

¹Dies mag sich mittlerweile geändert haben. Die Nicht-Erwähnung ähnlicher Anleitungen bedeutet

1 Einleitung

Webseiten und in diversen Foren, doch auf konkrete Probleme wird nie eingegangen. Alternativ zu diesem Skript kann man auch eine Offline-Installation durchführen². Die Offline-Installation hat Vor- und Nachteile. Ein klarer Vorteil ist, dass man das System direkt „in der Hand“ hat, und somit Fehler vor Ort beheben kann. Nachteil ist, dass man eine zum Server baugleiche Festplatte braucht und dass man das Image per Standleitung und dem Unix-Befehl `dd`³ installieren muss. Es ist sicherlich eine Alternative – man möge selbst entscheiden, welche Methode man wählt.

1.3 Voraussetzungen

Warnung! Wer nicht sicher im Umgang mit FreeBSD ist, ist hier leider am falschen Ort. Obwohl die Autoren versucht haben, dieses Dokument so einfach wie möglich zu halten, ist es zwingend notwendig, Erfahrung mit FreeBSD (oder zumindest mit UNIX-Derivaten) zu haben.

Die Voraussetzungen für den Einsatz des Depenguinators auf einem STRATO-Server sind neben dem fertig eingerichteten Server selbst ein Root-Zugang zu einer Linux-Box. Im Extremfall kann man auch das vorinstallierte Linux-System nutzen, das auf dem Server haust. Hierzu ist jedoch eine Neukompilation des Linux-Kernels notwendig, da benötigte Treiber in der Regel fehlen⁴. Eine Anleitung, wie man das bewerkstelligt, geht über den Rahmen dieses How-Tos hinaus.

nicht, dass wir dieses How-To für besser halten.

²bsdboys.net haben eine Anleitung geschrieben, die interessant für diejenigen ist, die es offline versuchen wollen.

³<http://web.cse.msu.edu/cgi-bin/man2html?dd?1m?usr/man>

⁴Etwa das Loop-Modul, um ISO-Images ins Dateisystem zu mounten.

2 Vorbereitung

Bevor es richtig losgeht, sollte im Konfigurationsmenü¹ die serielle Konsole aktiviert werden. Eine Neuinstallation der seriellen Konsole kann bis zu einer Stunde andauern, und statt die Stunde nachher „abzuwarten“, können wir die Konsole nun „bestellen“ und in der Zeit schon einmal loslegen.

Serielle Konsole bestellt? Dann wird das Rescue-System gestartet – ein Hardreset sollte hier durchgeführt werden. Da wir das alte System nicht mehr benötigen, brauchen wir uns keine Gedanken über eine unsaubere Root-Partition zu machen. Sämtliche Informationen, die wir benötigen, bekommen wir aus dem Rescue-System.

Rescue-System gestartet und eingeloggt? Dann geht es weiter...

2.1 Sammeln von Informationen

Hier darf man nicht ungeduldig werden. Das Sammeln der Informationen ist wichtig, und es darf kein Fehler gemacht werden. Bitte sorgfältig und gewissenhaft vorgehen.

Als erstes sollten Netzwerkinformationen erfasst werden. Dies geschieht mit Hilfe von `ifconfig`. Folgende Informationen sind wichtig:

- IP-Adresse des Servers,
- MAC-Adresse der Netzwerkkarte des Servers,
- IP-Adresse des verwendeten Nameservers und
- IP-Adresse des Gateways.

2.2 Konfiguration des Depenguinators

Auf der heimischen Linux-Box² wird nun der Depenguinator heruntergeladen und konfiguriert. Der Depenguinator erstellt aus einer Handvoll Konfigurationsdateien und einem CD-Image von FreeBSD 5.2.1 ein anderes Image, das nachher per `dd` auf die Server-Festplatte installiert wird.

¹STRATO-Konfigurationsmenü: <http://config.strato.de>

²Der Depenguinator erstellt keine gültigen Images mit FreeBSD.

2 Vorbereitung

Zuerst wird das Depenguinator-Image heruntergeladen:

```
http://www.daemonology.net/depenguin/depenguin-1.1.tar.gz  
oder  
http://strato.alpha-labs.net/depenguin-1.1.tar.gz
```

Die Autoren haben Beastie erfolgreich mit der Version 1.1 installiert. Höhere Versionen können laufen, sind aber ungetestet. Nach dem Entpacken sollte ein neues Verzeichnis mit folgendem Inhalt existieren:

```
total 490  
drwx--x--x  2 chris  chris   512B Oct 25 14:34 .  
drwxrwxr-x  4 chris  chris   512B Oct 25 14:34 ..  
-rw-----  1 chris  chris   1.1K Feb 27  2004 initconf  
-rw-----  1 chris  chris   1.3K Dec 26  2003 initconf.conf.dist  
-rw-----  1 chris  chris   8.0K Dec 26  2003 label.40M  
-rw-----  1 chris  chris    89B Dec 26  2003 loader.conf  
-rw-----  1 chris  chris  433K Dec 26  2003 makefs-20031226.tar.gz  
-rw-----  1 chris  chris   2.3K Dec 27  2003 makeimage.sh  
-rw-----  1 chris  chris   382B Dec 26  2003 mdinit  
-rw-----  1 chris  chris    45B Dec 26  2003 mdinit.conf
```

Es sollten alle Dateien von Haus aus richtig konfiguriert sein, bis auf die `initconf.conf.dist` und die `initconf`. Diese Dateien schauen wir uns gleich an.

Wichtig! Jetzt die Datei `initconf.conf.dist` nach `initconf.conf` kopieren.

Widmen wir uns nun der `initconf.conf`. Diese wird mit eurem Lieblings-Texteditor bearbeitet und mit den oben gesammelten Daten befüllt.

Achtung! Die neuen Server von STRATO brauchen Netmask 255.255.0.0 und nicht mehr 255.255.255.0. Schaut nach, ob euer Default-Gateway noch in eurem Netz liegt. Falls nicht, Netmask erhöhen!

Ein Beispiel gibt es in Kapitel B.1 auf Seite 31.

Desweiteren muss noch die serielle Konsole auf 57.600 Baud gesetzt werden. Dies geschieht durch das Editieren der `initconf`-Datei. Wir fügen folgende Zeile hinzu:

```
echo "ttyd0 \"/usr/libexec/getty std.57600\" vt100 on secure" >/etc/ttys
```

Und zwar über diesem Block:

```
for interface in ${initconf_interfaces}; do
  macaddr='eval echo "\\$initconf_mac_${interface}"'
  ipaddr='eval echo "\\$initconf_ip_${interface}"'
  netmask='eval echo "\\$initconf_netmask_${interface}"'
  echo ${ipaddr} ${initconf_hostname} >> /etc/hosts
  echo ${ipaddr} ${initconf_hostname}. >> /etc/hosts
```

und unter dieser Zeile:

```
echo 127.0.0.1 localhost > /etc/hosts
```

Diese Informationen sind wichtig! Dreifach überprüfen! Nachdem diese beiden Dateien sorgfältig editiert wurden, ist es an der Zeit, ein FreeBSD-Image zu besorgen. Wichtig ist dabei, dass es mindestens 5.x sein muss. Ein 4.x- oder älteres Image wird nicht funktionieren. Die Autoren konnten nur mit dem 5.2.1-Image³ einen Erfolg verzeichnen⁴:

<ftp://ftp-archive.freebsd.org/pub/FreeBSD-Archive/old-releases/i386/ISO-IMAGES/5.2.1/>

Gebraucht wird nur die erste CD-ROM⁵ der das Miniinst-Image. Das ISO-Image einfach in das Depenguinators-Verzeichnis verschieben.

Nochmals alles kontrollieren. Die meisten „Es bootet nicht“-Probleme resultieren aus schlechter Konfiguration in genau diesem Schritt.

Ein Debuggen ist nicht möglich, denn man hat ja noch keine Möglichkeit, auf den Server zuzugreifen.

2.3 Erstellen des Depenguinators-Images

Nachdem alles richtig konfiguriert ist, können wir nun mit dem Bau des Depenguinators beginnen. Dieser Schritt ist weitgehend automatisiert – leider ist er aber auch nicht wirklich einsichtig, da der Depenguinator auch bei Fehlern nicht abbricht, einfach weitermacht und so ein fehlerhaftes Image erzeugt⁶. Die Hauptursache dafür ist das Fehlen des loop-Moduls oder des iso9660-Moduls im Linux-Kernel. Diese Module fehlen z.B. auf (gescheit) konfigurierten Servern, wo man solche Module eigentlich nicht braucht.

³5.2.1 ist eine alte Version. Das ist aber kein Problem, denn das endgültige FreeBSD-System ist problemlos auf eine aktuelle Version zu bringen.

⁴Wir haben nun definitive Bestätigung, dass es mit 5.3 oder höher nicht geht.

⁵o

⁶Der Depenguinator erstellt sogar ein Image, ohne dass er die FreeBSD-CD mounten kann. Das resultierende Image ist genauso groß wie ein geglücktes Image.

2 Vorbereitung

Am besten vorher einmal testen, ob diese Module auch einwandfrei funktionieren. Als root:

```
# mkdir mnt
# mount -o loop -t iso9660 5.2.1-RELEASE-i386-disc1.iso mnt
# ls mnt/
[...]
# umount mnt
# rmdir mnt
```

Sollte der `ls`-Befehl kein leeres Verzeichnis angezeigt haben, so funktioniert hier alles. Analog dazu: Wenn `ls` ein leeres Verzeichnis anzeigt, stimmt was nicht.

Wir gehen von einem Erfolg aus und generieren das Image:

```
$ ./makeimage.sh 5.2.1-RELEASE-i386-disc1.iso
```

beziehungsweise

```
$ ./makeimage.sh 5.2.1-RELEASE-i386-miniinst.iso
```

Nach einer Menge Text sollte die Datei `disk.img` erstellt worden sein. Diese ist – soweit richtig konfiguriert – ein Image, das eine Partitionstabelle, einen Bootsektor und eine FreeBSD-Partition enthält und etwa 40 MB groß ist. Es wird davon ausgegangen, dass dieses Image fehlerfrei ist, sprich: Die Konfiguration wurde ordnungsgemäß durchgeführt, und die ISO-Datei konnte gemounted werden. Sollte im nächsten Schritt der Server nicht booten, so liegt es an einem fehlerhaften Image! Vorsicht Falle: Das Image ist auch nach einem fehlerhaften Bau 40 MB gross!

3 Arbeiten mit dem Depenguinator

3.1 Konzept

Jetzt ist es soweit. Das Image wird auf dem Root-Server installiert. Dazu sollte dieser bereits im Rescue-Modus sein, und eine SSH-Verbindung als root bestehen.

Als erstes müssen wir Platz machen. Das Problem dabei ist, dass das Image erst auf den Server muss. Dieses Image überschreibt dann die ersten 80 MB der Festplatte. Deshalb muss besonderer Wert darauf gelegt werden, dass die ersten 80 MB frei sind! Darum kümmern wir uns im nächsten Schritt.

Es gibt eine alternative Methode, das Image auf den Server zu bekommen. Diese Methode wird auf Seite 13 beschrieben.

3.2 Partitionieren (Rescue Modus) (Linux)

Warnung! Alle Daten gehen nun verloren! Letzte Chance!

`fdisk /dev/hda` starten und alle Partitionen löschen. Minimalistische Einführung in `fdisk`: Nach dem Start erscheint (in etwa)¹ folgendes:

```
Disk /dev/hda: 30.7 GB, 30743838720 bytes
16 heads, 63 sectors/track, 59570 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	40	20128+	83	Linux
/dev/hda2		41	2025	1000440	82	Linux swap
/dev/hda3		2026	59570	29002680	83	Linux

Command (m for help):

Die wichtigsten Befehle sind:

¹Dieser Befehl wurde auf dem heimischen Computer ausgeführt, daher bitte keine Panik bekommen wenn die Zahlen abweichen.

3 Arbeiten mit dem Depenguinator

p - zeigt alle Partitionen an
d - löscht Partitionen
n - erstellt Partitionen
t - setzt Partitionstyp.
w - speichern & quit
q - quit

Nachdem alle Partitionen gelöscht sind, wird eine neue Partition erstellt. Diese beginnt so ziemlich am Ende der Festplatte; meine Festplatte hatte etwa 7900 Zylinder, und meine Partition ging von Zylinder 7000 bis 7900. Wichtig ist auch, dass der Typ auf 83 gesetzt wird. Eigentlich muss man nur gesunden Menschenverstand benutzen, also: Partition weit hinten erstellen und nicht zu klein machen. Hier kann eigentlich kaum etwas schiefgehen.

Partition erstellt? Speichern und beenden.

Nun muss die Partition formatiert:

```
# mke2fs /dev/hda1
```

und gemountet werden:

```
# cd /  
# mkdir bsd  
# mount -t ext2 /dev/hda1 bsd/
```

Jetzt haben wir eine Partition, die sich am Ende der Festplatte des Servers befindet. Sie ist formatiert und gemountet. Nun können wir vom heimischen PC die Datei `disk.img` auf die neue Partition installieren.

3.3 Installation des Depenguinators

Von zu Hause:

```
# scp disk.img root@meinstratoserver.com:/bsd/
```

Dies funktioniert natürlich nur aus dem Verzeichnis heraus, in dem auch der Depenguinator liegt. Das Hochladen dauert selbst bei DSL-Leitungen noch eine Weile. Deshalb ist nun die ideale Zeit für eine Kaffeepause²!

Nach dem Kopieren noch einmal die Integrität der Dateien überprüfen:

²Wir empfehlen SHB von Coffee-Nation.

Auf dem heimischen Rechner:

```
# md5sum disk.img  
ea8ad43f438f63e74502a686daafc037  disk.img
```

Auf dem Root-Server:

```
# md5sum /bsd/disk.img  
ea8ad43f438f63e74502a686daafc037  disk.img
```

Wenn die MD5-Summen nicht übereinstimmen, ist das Image fehlerhaft und muss nochmals übertragen werden! Wichtig: Du hast ein eigenes Unikat³ dieser MD5-Summe, diese stimmt auf keinen Fall mit der obigen überein! Was zählt ist, dass deine zwei identisch sind.

Bei einer korrekten MD5-Summe können wir nun das Image auf die Festplatte installieren. Da es sich hierbei um einen Sector-by-Sector-Kopiervorgang handelt, kann es schonmal etwas länger dauern.

Vorher nochmal überprüfen, ob man sich auf dem richtigen Rechner befindet, sonst ärgert man sich grün und blau!

```
# dd if=/bsd/disk.img of=/dev/hda  
# sync
```

3.4 Alternative Methode

Man kann die bisher genannten Schritte in diesem Kapitel überspringen, indem man von Zuhause das Image direkt auf die Server-Partition schreibt. Die Daten werden also ohne Umwege über Buffering auf die Server-Festplatte geschrieben.

Dies geschieht von daheim mit dem Befehl:

```
# cat disk.img | ssh root@meinstratoserver.com "dd of=/dev/hda"
```

Dies ist mit Vorsicht zu genießen, denn es ist auf keinen Fall so sicher wie die erste Methode, denn:

- Es können Übertragungsfehler auftreten. Da das Image nicht zwischengespeichert wird, sondern direkt auf die Partition geschrieben wird, sind Übertragungsfehler nicht zu entdecken.

³An die Theoretiker: Wir gehen von einer kollisionsfreien Hashfunktion aus.

3 Arbeiten mit dem Depenguinator

- Die Verbindung zum Root-Server kann jederzeit abbrechen. Sollte dies geschehen, darf man wieder von vorne mit der Übertragung beginnen.
- Es entfällt die Möglichkeit, die Integrität der Image-Datei zunächst mit md5 zu überprüfen.

Welche Geschmacksrichtung man wählt, bleibt die Wahl des Administrators.

3.5 Noch eine Methode

Das Depenguinator-Image (disk.img) mit `scp` auf den Strato-Server kopieren, und anschließend einfach mit `dd if=disk.img of=/dev/hda` auf die Platte schreiben. Wenn man will, kann man vorher noch mit md5 die Prüfsumme der Kopie mit der des Originals vergleichen. Einzige Gefahr dabei ist, dass das Image in den ersten 40 MB der Platte liegt und sich somit selber überschreiben würde. Die Wahrscheinlichkeit dafür ist sehr gering, aber vorhanden. Hier tauscht man Sicherheit gegen einen Zeitvorteil ein.

3.6 Neu starten

Somit wäre das Image auf der Platte. Im Konfigurationsmenü von Strato stellt man nun den Server wieder auf „Normaler Boot mit Hardwarereset“. Nach 5 bis 10 Minuten macht es *pling*, und der Server startet neu.

Kleine Anmerkung: Als ich das gemacht habe, habe ich in der Zeit eine Pizza gemacht und gegessen. Als ich fertig war, kam der erste Ping an.

Beten.

Der Server kann schon mal 5-15 Minuten brauchen zum Booten. Also keine Panik. Außerdem ist zu beachten, dass der Server (eventuell) noch kein Netzwerk hat. Ein Anpingen des Servers ist hier nur bei absolut korrekter Konfiguration des Servers möglich. Die einzige Methode, bei Problemen dennoch auf den Rechner zu kommen, ist die serielle Konsole, auf die wir uns nun einloggen.

Wer nur die Meldung

```
PXELINUX 1.76 2002-08-27 Copyright (C) 1994-2002 H. Peter Anvin
Booting from local disk...
```

sieht (und sonst nichts weiter) leidet an fehlerhaft konfigurierter serieller Konsole. Vermutlich ist die `/etc/ttyS` nicht korrekt übernommen worden. Diese Datei sollte in diesem Schritt nur eine einzelne Zeile beinhalten:⁴

⁴bzw. mehr, je nachdem ob Headless oder nicht.

```
echo "ttyd0 \"/usr/libexec/getty std.57600\" vt100 on secure" >/etc/ttys
```

Dies sollte der Depenguinator allerdings selbst erledigen. Wenn nicht, habt ihr den entsprechenden Eintrag in der `initconf` vergessen oder dabei einen Fehler gemacht. Sollte es trotzdem nicht funktionieren, ist VMware⁵ sehr hilfreich. Die Demo läuft 30 Tage – genug Zeit zum Debuggen. In VMware einfach eine Disk erstellen, von CD booten und dann wie auf dem STRATO-Server das Image per `scp` auf den virtuellen Rechner kopieren und mittels `dd` auf die Festplatte schreiben. Wenn VMware dann bootet, könnt Ihr sehen, woran es hapert.

Wichtig! Wenn Ihr einen emulierten PC oder keinen Headless-Server depenguiniert, solltet ihr statt ausschließlich `ttyd` (serielle Konsole) natürlich zusätzlich die normalen Konsolen zur Verfügung stellen.

Falls ihr die serielle Konsole doch von Hand aktivieren müsst, bearbeitet ihr die Datei `/etc/ttys` und ändert

```
#ttyd0 \"/usr/libexec/getty std.57600\" vt100 off secure
```

in

```
ttyd0 \"/usr/libexec/getty std.57600\" vt100 on secure
```

Bei einigen FreeBSD-Versionen ist die Zeile mit einer Raute auskommentiert, bei anderen ist `ttyd0` lediglich auf „off“ gesetzt.

Nach viel Zittern sollte da folgendes stehen:

```
FreeBSD/i386 (meinstratoserver) (ttyd0)
```

```
login:
```

Herzlichen Glückwunsch! Und weiter geht's...

⁵<http://vmware.com> - ein kommerzieller PC Emulator

3 Arbeiten mit dem Depenguator

4 Unter FreeBSD

In diesem Kapitel werden wir zunächst das Netzwerk bereitstellen, die Festplatten partitionieren, das System installieren und die notwendige Konfiguration durchführen.

4.1 Netzwerk (vorerst) einrichten

Eingeloggt als `root` machen wir erstmal das Netzwerk startklar. Das ist ziemlich einfach:

```
# dhclient fxp0
# ifconfig fxp0 netmask 255.255.255.0
- bzw -
# ifconfig fxp0 netmask 255.255.0.0
# route add default [IP des Gateways]
# echo "nameserver [IP des Nameservers]" >/etc/resolv.conf
```

Einfach die zwei Platzhalter gegen die echten IPs tauschen (die Adressen hatten wir ja vorhin aufgeschrieben). Ein ping vom heimischen PC auf die IP des Servers sollte nun funktionieren, genau wie ein ping vom Server auf einen Internet-Rechner, wie `web.de`:

```
# ping web.de
PING web.de (217.72.195.42): 56 data bytes
64 bytes from 217.72.195.42: icmp_seq=0 ttl=51 time=18.953 ms
64 bytes from 217.72.195.42: icmp_seq=1 ttl=51 time=14.679 ms
64 bytes from 217.72.195.42: icmp_seq=2 ttl=51 time=15.793 ms
```

Bitte nun per SSH regulär auf dem Server einloggen und die serielle Konsole schließen. Das hat folgenden Grund: Wir benutzen gleich `sysinstall` und das verwendet Farbausgabe. Farbe auf der seriellen Konsole im aktuellen Systemzustand ergibt zwar lustige, aber wenig hilfreiche Informationen.

4.2 Partitionieren

Bisher läuft FreeBSD aus dem RAM heraus. Wir haben ein funktionierendes Netzwerk, und sind per SSH als `root` auf dem Server eingeloggt.

4 Unter FreeBSD

Da wir aus einem BSD heraus ein BSD installieren, müssen wir einiges beachten. Zum Beispiel können wir nicht in das Root-Verzeichnis installieren, sondern müssen ein anderes Verzeichnis auswählen.

```
# mkdir /hda
```

Das ist unser Mountpoint und Installroot. `sysinstall` starten, im Menü *Configure* auswählen, dann *Fdisk*. Wir wollen die ganze Festplatte als eine FreeBSD-Slice verwenden, daher können wir hier ruhig die Auto-Funktion nehmen. Im Gegensatz zu dem, was `sysinstall` uns vorschlägt, wollen wir die Partitionstabelle speichern und *Fdisk* beenden. Bei der Frage, ob wir einen Boot-Manager installieren wollen, antworten wir mit

```
BootMgr Install the FreeBSD Boot Manager.
```

Da `sysinstall` bei vielen Installationen dieser Art Fehler nach dem Partitionieren produziert, müssen wir `sysinstall` an dieser Stelle erst einmal verlassen und dannach wieder starten. Ansonsten klappt das Partitionieren nicht. Im Hauptmenü wählen wir nun unter *Configure* den Eintrag *Label* aus. Hier partitionieren wir unsere Festplatte manuell – oder so wie es jeder mag. Wichtig ist nur, dass die Slices als Mount-Prefix `/hda` bekommen, sonst mountet `sysinstall` das neue Filesystem nach `/` und überschreibt somit die Ramdisk, was einem Neuanfang gleichkommt¹.

Die Autoren haben ihre Slices so partitioniert:

```
FreeBSD Disklabel Editor

Disk: ad0          Partition name: ad0s1   Free: 0 blocks (OMB)

Part      Mount          Size Newfs   Part      Mount          Size Newfs
----      -
ad0s1a    /hda             57500MB *
ad0s1b    swap            1143MB SWAP

[...]
```

The following commands are valid here (upper or lower case):

```
C = Create          D = Delete      M = Mount pt.      W = Write
N = Newfs Opts     Q = Finish     S = Toggle SoftUpdates  Z = Custom Newfs
T = Toggle Newfs   U = Undo       A = Auto Defaults    R = Delete+Merge
```

Use F1 or ? to get more help, arrow keys to select.

¹Ihr erinnert euch sicherlich: Rescue-Modus, Disk-Image auf den Server kopieren und Disk-Image auf die Server-Platte schreiben.

Partitionen eingetragen? Richtige Mountpoints? Sicher? Dann speichern und beenden. Wenn alles gut gelaufen ist, sollte mount die neuen Dateisysteme als gemountet kennzeichnen.

4.3 Konfiguration von sysinstall

Wieder in sysinstall, gehen wir direkt in die *Options*, und stellen das System wie folgt ein:

Options Editor

Name	Value	Name	Value
----	-----	----	-----
NFS Secure	NO	Install Root	/hda
NFS Slow	NO	Browser package	links
NFS TCP	NO	Browser Exec	/usr/local/bin/links
NFS version 3	YES	Media Type	FTP
Debugging	NO	Media Timeout	300
No Warnings	NO	Package Temp	/hda/systmp
Yes to All	NO	Newfs Args	-b 16384 -f 2048
DHCP	YES	Fixit Console	serial
IPv6	NO	Re-scan Devices	<*>
Skip PCCARD	YES	Use Defaults	[RESET!]
FTP username	ftp		
Editor	/usr/bin/ee		
Tape Blocksize	20		
Extract Detail	high		
Release Name	5.4-RELEASE		

Use SPACE to select/toggle an option, arrow keys to move, ? or F1 for more help. When you're done, type Q to Quit.

Hier noch ein Wort zur Release: Release 5.2.1 ist nun fast von allen FTP-Servern verschwunden. Man kann hier die 5.4-RELEASE angeben, was auch geht, nur gibt es in 5.4 das Crypto-Paket nicht. Das ist aber nicht weiter schlimm, denn der Inhalt wurde ins Base-System portiert. Also: Fehlermeldungen betreffs crypto einfach ignorieren, es geht auch so.

4.4 Installation

Das neue, gemountete Filesystem schreit förmlich nach einer Installation. Aber Vorsicht! sysinstall braucht zum Installieren temporären Platz, den es sich von /tmp herholt.

4 Unter FreeBSD

Problem: `/tmp` liegt innerhalb der Ramdisk, deshalb:

Innerhalb des neuen Dateisystems ein neues Verzeichnis anlegen, das die temporären Dateien beherbergt. `mkdir /hda/systmp2` und das war's. Dann `sysinstall` starten, und dreifach überprüfen, dass *Installroot* auf `/hda` und das Package-Temp auf `/hda/systmp` steht. Und wo wir schon dabei sind können wir auch den Media Type auf FTP stellen. Sucht euch einen Mirrorserver eurer Wahl aus, am Besten natürlich einen aus Deutschland.

Nachdem wir `sysinstall` konfiguriert, eine `/hda`-Partition und ein `/hda/systmp`-Verzeichnis haben, können wir nun das System installieren. Da wir wissen, was wir wollen, nehmen wir den Pfad der Custom-Installation.

Optionen sollten eingestellt sein, partitioniert und gelabelt haben wir auch schon. Bleibt nur auszuwählen, was wir installieren wollen. Die Autoren empfehlen *Minimal*. Ein Installationsmedium sollte schon eingestellt sein – wenn nicht, macht es nun.

Alles fertig? Dann installieren. Viel Glück!

Nach der Installation wird man gefragt, ob man noch weitere Optionen einstellen will, wie z.B. Zeit- oder Konsoleinstellungen. Diese können wir erstmal auslassen, da `sysinstall` diese nach `/etc` speichert und nicht nach `/hda/etc`.

²nicht `/hda/tmp`, denn das wird später das `tmp`-Verzeichnis des Systems!

5 Konfiguration des installierten Systems

Mit einem frischen FreeBSD-System auf der Platte will man gleich neustarten und loslegen. Fatal! Es muss noch einiges gemacht werden. Aber erstmal beenden wir `sysinstall`. Auf der Konsole geht es weiter.

5.1 chroot

Als erstes begeben wir uns in das neue System und setzen so `/hda` nach `/`. Das heißt: Alles was nun kommt, ist *absolut* und nicht relativ zum Installations-Pfad!

Mit `chroot /hda` begeben wir uns ins System.

5.2 Sendmail

Sendmail wird mitinstalliert und startet auch automatisch. Leider hält dies den Bootvorgang unnötig auf, denn sendmail hat noch keinen gültigen Domain-Namen. Daher deaktivieren wir Sendmail vorerst. Wir editieren die `/etc/rc.conf` und fügen hinzu:

```
sendmail_enable="NONE"
```

5.3 /etc/rc.conf

Diese Datei konfiguriert und steuert elementare Dinge, daher ist eine Konfiguration vorab unerlässlich. Gewisse Dienste sollten bei einem Neustart auch gestartet werden, zum Beispiel `sshd`. Das sollte bei Euch stehen:

```
fsck_y_enable="YES"
background_fsck="NO"
inetd_enable="YES"
sshd_enable="YES"
portmap_enable="NO"
```

```
hostname="VanGoughderFassenson"
network_interfaces="lo0 fxp0"
```

5 Konfiguration des installierten Systems

```
ifconfig_fxp0="inet 81.169.181.abx netmask 255.255.255.0"  
defaultrouter="81.169.185.1"
```

Mit Ausnahme der letzten 4 Zeilen, die solltet ihr schon anpassen. Beachtet bitte, dass der Name des Netzwerk-Devices abhängig vom verwendeten Treiber ist. Mit anderen Worten: `fxp0` könnte auch `sis0` heißen, falls STRATO Boards mit anderen Ethernet-Chips verbaut.

5.4 /etc/fstab

Da `sysinstall` in der aktuellen Version nicht wirklich mit Non-root-Dateisystemen umgehen kann, installiert es die `fstab` nach `/etc` und nicht nach `/hda/etc` (oder wie auch immer das Verzeichnis bei euch heißt). Hier könnt Ihr entweder das chroot mit `Ctrl+D` verlassen und die von `sysinstall` installierte `/etc/fstab` von Hand nach `/hda/etc/fstab` kopieren oder eben eine eigene anlegen. Im Anhang findet Ihr eine Beispiel-`fstab` für Systeme mit zwei Partitionen (eine Swap- und eine UFS-Partition).

Ansonsten die von `sysinstall` generierte `fstab` nutzen, dazu gehen wir aus dem chroot raus und kopieren die Datei mit:

```
cp /etc/fstab /hda/etc/fstab
```

Jetzt noch die Datei überprüfen – das sollte es gewesen sein.

5.5 /etc/ttys

Hier liegt das gleiche Problem wie bei der `fstab` vor. Abhilfe, ausserhalb des chroots:

```
cp /etc/ttys /hda/etc/ttys
```

Jetzt noch die Datei editieren, und diese Zeile suchen:

```
ttyd0 "/usr/libexec/getty std.9600" dialup off secure
```

und durch

```
ttyd0 "/usr/libexec/getty std.57600" vt100 on secure
```

ersetzen. Dies gilt, wie immer, natürlich nur für Headless-Server. Sollte eure Serverkonfiguration anders aussehen, so müsst Ihr hier selber dafür Sorge tragen.

5.6 /etc/sysctl.conf

Wieder im chroot wird /etc/sysctl.conf um diese Zeile bereichert:

```
machdep.conspeek=57600
```

5.7 /etc/resolv.conf

Nun die Datei /etc/resolv.conf anlegen, und Nameserver eintragen, etwa so:

```
search serverkompetenz.net
nameserver 81.169.163.106
nameserver 81.169.163.104
```

Wer das nicht macht, kann sich nicht mehr per SSH einloggen, denn SSH will ja ein reverse-lookup (welchen man auch ausschalten kann – aber wozu Symptome beheben wenn man direkt die Krankheit heilen kann?)

5.8 Root-Passwort

Der Root-Account hat noch kein Passwort, was ein Einloggen per SSH unmöglich macht. passwd ausführen und ein Passwort setzen. Und bitte, tut euch den Gefallen und wählt ein sicheres Root-Passwort¹.

5.9 Notfall-Hintertür

Sollte beim ersten Neustart sshd nicht starten und die serielle Konsole nicht funktionieren, brauchen wir eine Hintertür. Einer der Autoren hatte ein fertiges System, konnte es anpingen, aber der sshd war nicht gestartet. Resultat: Alles nochmal von vorn.

Wir bauen uns eine Hintertür: Mit sysinstall legen wir einen User an, den wir in die wheel-Gruppe setzen. Anschliessend aktivieren wir den telnetd. Einen User anzulegen sollte jeder FreeBSD-User können, und der telnetd wird in der /etc/inetd.conf aktiviert, indem man die telnet-Zeile findet und die Raute entfernt. Nicht vergessen, auch für diesen User ein Passwort zu setzen, damit dieser sich einloggen kann.

Da wir nun auf telnetd (vorübergehend) angewiesen sind, müssen wir diesen auch noch aktivieren. Schreibt in die /etc/rc.conf:

```
inetd_enable="yes"
```

¹Password-Advisory: <http://www.cs.umd.edu/faq/Passwords.shtml>

5.10 Neustart

Alles doppelt überprüft? Dann die chroot-Sitzung beenden (`logout` oder `Ctrl+D`) und die Partition unmounten (`umount /hda`). Nochmal mit `mount` überprüfen, ob nichts mehr (außer der Ramdisk) gemountet ist, dann allen Mut zusammenfassen und neustarten.

Bitte beachten: Der Boot-Loader ist noch nicht sichtbar. Dazu müssen wir das System neu kompilieren – das erledigen wir im nächsten Schritt.

Ausserdem erwartet FreeBSD in einigen Versionen eine zufällige Eingabe, um den SSH-Key zu generieren. Diese erwartete Eingabe hat einen 5-Minuten-Timeout. Kurzum: Da es keine Möglichkeit gibt, diese Eingabe vorzunehmen, dauert der Bootprozess an dieser Stelle 5 Minuten länger.

6 Das neue System

Nach dem hoffentlich gelungenen Reboot hat man das neue System vor sich. Die serielle Konsole sollte laufen. Falls nicht, kann man sich per SSH einloggen. Sollte auch das nicht funktionieren, kann man auf die Notfall-Hintertür zurückgreifen: Telnnet als Backdoor-User, und dann mittels `su` auf den Root-User wechseln.

Doch bevor Ihr Euch an die Konfiguration des Systems begeben, gibt es noch ein paar kleine Dinge, die es zu erledigen gilt, um wirklich in den vollen Genuss der Konsole zu kommen.

6.1 cvsup

Dazu müssen wir zunächst ein `cvsup` durchführen. Hat man `cvsup` nicht installiert (wegen der Minimalinstallation), so kann man nun `sysinstall` aufrufen und nachträglich das Paket `cvsup-without-gui` installieren, alternativ:

```
# pkg_add -r cvsup-without-gui
```

Ab hier ist alles wie zuhause. Wer will, kann die `/etc/cvsup-` bzw. `/etc/cvsup-ports-` Dateien aus dem Anhang übernehmen. Jetzt den FreeBSD-Source-Tree aktualisieren und das System auf den neuesten Stand bringen:

```
root@pa:~# cvsup /etc/cvsup  
[...]  
root@pa:~# cd /usr/src/sys/i386/conf
```

Hier nun eine eigene Kernel-Konfigurationsdatei anlegen und editieren. Wichtig:

```
options CONSPEED=57600
```

als Option eintragen. Nun noch den Standard-Namen und Parameter für die serielle Konsole in `/etc/make.conf` eintragen:

```
KERNCONF=EuerConfigNameHier  
BOOT_COMCONSOLE_SPEED=57600
```

6.2 /boot.config

Um das Bootmenü sehen zu können, muss im Root-Verzeichnis eine Datei angelegt werden, die `boot.config` heißt. In diese tragen wir ein:

```
-h
```

6.3 World, Kernel und Loader neu bauen

Hier ist es genauso wie zu Hause:

```
cd /usr/src && make -s buildworld buildkernel
```

Das kann schon etwas dauern. Kaffepause. Wenn das System fertig ist, und keine Fehler aufgetreten sind, seid ihr quasi fertig – bis auf die Tatsache, dass das System noch installiert werden muss. Und vorher muss man das „alte“ System noch auf FreeBSD 5.4 oder 6.1 upgraden:

```
# mergemaster -p  
# mergemaster -ai
```

Das fügt dem System noch die fehlenden User hinzu. Man soll zwar in den Single-User-Mode gehen um `mergemaster -ai` auszuführen, aber uns sind keine Probleme in dieser Hinsicht bekannt – ist also eher Geschmackssache.

Besonders wichtig ist `mergemaster -ai` bei einem Upgrade auf 6.1 - es werden wichtige `rc.d`-Skripte aktualisiert!

Vorsicht! Auf einem Production-Server, auf dem vielleicht 50 Dienste gestartet und zehn User gleichzeitig eingeloggt sind, empfehlen die Autoren doch lieber den Single-User-Mode. Aber in diesem Fall ist ja hoffentlich niemand außer uns auf dem System (und es sind auch keine Dienste gestartet) – was effektiv dem Single-User-Mode gleichkommt.

Nun sollte noch der Boot-Loader installiert werden:

```
# cd /sys/boot  
# make depend all install
```

Nun hat man zwar den richtigen Boot-Block gebaut und in `/boot` liegen, nur muss dieser auch an der richtigen Stelle auf der Festplatte geschrieben werden. `bsdlabel` kann uns hier gute Dienste leisten. Wichtig ist, daß man die Slice angibt, von welcher man bootet, nicht die Festplatte oder die Partition. Beispiel:

```
# bsdlabel -B /dev/ad0s1
```

6.4 Der letzte Neustart

Nochmals überprüfen, ob alle Konfigurationsdateien editiert sind:

1. `/etc/make.conf` (`BOOT_COMCONSOLE_SPEED=57600`)
2. `/etc/sysctl.conf` (`machdep.conspeed=57600`)
3. `/etc/ttys` (`ttyd0`)
4. `/boot.config` (`-h`)

6.5 Backdoor entfernen

Die vorhin eingebaute Backdoor sollte nun aus Sicherheitsgründen entfernt werden. Editiert dazu wieder die `/etc/inetd.conf` und kommentiert die `telnetd`-Zeile wieder aus. Anschliessend sollte noch der Benutzer, den Ihr angelegt habt, gelöscht werden.

6.6 Letzte Worte

Nun solltet Ihr ein neues FreeBSD-System haben, das den Boot-Loader auf der seriellen Konsole anzeigt. Das Netzwerk sollte laufen und alle Basis-Komponenten sollten einsatzbereit sein.

Ein letzter Reboot und ein abschließendes Gebet...

Nun sollte alles gehen, und Ihr könnt euch der Konfiguration eures Systems widmen.

Wir wünschen viel Spaß und Erfolg!

6 *Das neue System*

A Frequently Asked Questions

Es wurden kaum Fragen gestellt, daher ist diese Kategorie ziemlich leer. Falls Ihr Problem begegnet seid, kann eine eMail an die Autoren (siehe Deckblatt) helfen.

Q: Spielt die Partitionsnummer (1-4) eine Rolle bei der Einrichtung mit fdisk? (Kapitel: Partitionieren (Rescue Modus) (Linux))

A: Nein. Das wichtigste ist der Startsektor, der muss stimmen.

Q: Wo bekomme ich denn ein FreeBSD-5.2.1-Image her?

A: <ftp://ftp-archive.freebsd.org/pub/FreeBSD-Archive/old-releases/i386/ISO-IMAGES/5.2.1/>

Q: Kann ich auch das Miniinst nehmen?

A: Ja, auch das Miniinst von 5.2.1 funktioniert.

A Frequently Asked Questions

B Konfigurationsdateien

Hier sind nochmal alle Dateien gelistet, die im Laufe der Konfiguration und Installation von Null auf BSD nötig sind. Dateien des Depenguinators sind mit (dp) gekennzeichnet. Alle anderen Dateien beziehen sich auf ein installiertes System.

B.1 `initconf.conf(.dist)` (dp)

```
# Everything which you should need to edit
# to create a boot image is in this file.

# initconf_hostname corresponds to "hostname"
# in /etc/rc.conf
initconf_hostname="serverxyz.serverkompetenz.de"

# initconf_nameserver corresponds to the nameserver
# entry in /etc/resolv.conf
initconf_nameserver="81.169.163.106"

# initconf_gateway corresponds to the "defaultrouter"
# entry in /etc/rc.conf
initconf_gateway="81.169.185.1"

# initconf_rootpw is the root password which will
# be set. Root login will be enabled in sshd, so
# this allows you to login and install FreeBSD
# remotely.
# Make sure you only use alphanumeric characters
# here; other symbols may cause problems.
# Note that this is stored in plaintext! Don't
# use an important password for this.
initconf_rootpw='foomatic'

# These correspond to the ifconfig_* entries in
# /etc/rc.conf. We can't use the "real" names of
# interfaces, because they depend upon the order
# in which devices are detected and initialized.
```

B Konfigurationsdateien

```
# initconf_interfaces should be a list of
# names of interfaces.
initconf_interfaces="external"

# initconf_mac_F00 should be the MAC address
# of interface F00
initconf_mac_external="01:23:45:67:89:ab"

# initconf_ip_F00 should be the IP address
# of interface F00
initconf_ip_external="81.169.123.123"

# initconf_netmask_F00 should be the
# netmask of interface F00
initconf_netmask_external="255.255.255.0"
```

B.2 initconf (dp)

```
# KEYWORD: FreeBSD

. /etc/rc.subr

name="initconf"
start_cmd="initconf_start"
stop_cmd=":"

initconf_start()
{
    echo sshd_enable="YES" >> /etc/rc.conf
    echo PermitRootLogin yes >> /etc/ssh/sshd_config
    echo hostname="\${initconf_hostname}" >> /etc/rc.conf
    echo defaultrouter="\${initconf_gateway}" >> /etc/rc.conf
    echo domain . > /etc/resolv.conf
    echo nameserver ${initconf_nameserver} >> /etc/resolv.conf
    echo 127.0.0.1 localhost > /etc/hosts
    echo "ttyd0 \"/usr/libexec/getty std.57600" vt100 on secure" >/etc/ttys

    for interface in ${initconf_interfaces}; do
        macaddr='eval echo "\\${initconf_mac_${interface}}"'
        ipaddr='eval echo "\\${initconf_ip_${interface}}"'
        netmask='eval echo "\\${initconf_netmask_${interface}}"'
    done
}
```

```

echo ${ipaddr} ${initconf_hostname} >> /etc/hosts
echo ${ipaddr} ${initconf_hostname}. >> /etc/hosts
echo "ttyd0 \"/usr/libexec/getty std.57600\" vt100 on secure" >/etc/ttys

for real_if in `ifconfig -l`; do
    if ifconfig ${real_if} | grep -qi ${macaddr}; then
echo ifconfig_${real_if}=\`inet ${ipaddr} netmask ${netmask}\` >> /etc/rc.conf
        fi
    done
done

echo ${initconf_rootpw} | pw usermod root -h 0
}

load_rc_config $name
run_rc_command "$1"

```

B.3 /etc/cvsup

```

*default host=cvsup.de.FreeBSD.org
*default base=/usr
*default prefix=/usr

*default release=cvs tag=RELENG_5_4
(-- oder --)
*default release=cvs tag=RELENG_6_1

*default delete use-rel-suffix
*default compress
src-all

```

B.4 /etc/cvsup-ports

```

*default host=cvsup.de.FreeBSD.org
*default base=/usr
*default prefix=/usr
*default release=cvs tag=.
*default delete use-rel-suffix
*default compress
ports-all

```

B.5 /etc/make.conf

```
[...]  
BOOT_COMCONSOLE_SPEED=57600  
WITHOUT_X11=yes  
[...]
```

B.6 /etc/ttys

```
[...]  
# Dumb console  
ttyd0 "/usr/libexec/getty std.57600" vt100 on secure  
[...]
```

B.7 /etc/sysctl.conf

```
[...]  
machdep.conspeed=57600  
[...]
```

B.8 /etc/fstab

/dev/ad0s1b	none	swap	sw	0	0	
/dev/ad0s1a	/	ufs	rw,userquota		1	1